Session Types and Game Semantics
Synchrony and Asynchrony

Simon Castellan[1], Pierre Clairambault[2], Nobuko Yoshida[1]

[1]Imperial College London [2]LIP, ENS Lyon

14 april 2018
GaLoP 2018

The π-calculus [MPW92]

The π-calculus describes agents communicating through **channels**:

$$
\begin{aligned}
P, Q ::= & \, 0 \\
& \mid (P \mid Q) \\
& \mid (\nu ab)P && \text{restriction} \\
& \mid \bar{a}!\ell\langle u\rangle.\, P && \text{output} \\
& \mid \sum_{i \in I} a?\ell(x_i).P_i && \text{input} \\
& \mid P + Q && \text{nondet. choice}
\end{aligned}
$$

Communication: data (ℓ) and channels (u).

Short-hands: $\bar{a}\langle u\rangle := \bar{a}! \star \langle \vec{u}\rangle \qquad a(x) := a? \star (x)$

Game semantics for the π-calculus

Existing models:

- Laird [Lai05] refined by Tsukada & Sakayori [ST17] (for the **asynchronous** fragment)
- Hirschowitz *et. al.* [EHS15]

$\rightsquigarrow$ In this talk, focus on analyzing the first line of interpretation.

Game semantics for the π-calculus

Existing models:

- Laird [Lai05] refined by Tsukada & Sakayori [ST17] (for the **asynchronous** fragment)
- Hirschowitz *et. al.* [EHS15]

$\rightsquigarrow$ In this talk, focus on analyzing the first line of interpretation.

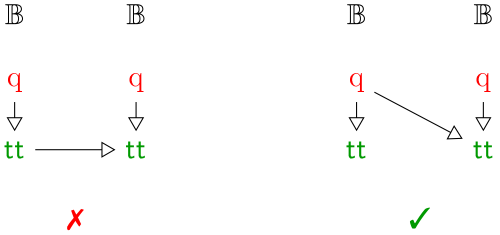
Basic idea: interpret channels as an effect like references:

$$[\![\sharp T]\!] = [\![T]\!]^{\perp} \times [\![T]\!]$$
$$[\![(\nu a)P]\!] = [\![P]\!] \odot \mathsf{cc}$$

Asynchrony: game semantics

Concurrent game semantics is tradtionally **asynchronous**:

$$\mathfrak{c}_A \odot \sigma = \sigma \Longrightarrow \sigma \text{ courteous [MM07, RW11]}$$



This forces some equations in the model:

$$[\![\bar{a}\langle u\rangle.\bar{b}\langle v\rangle.P]\!] = [\![\bar{b}\langle v\rangle.\bar{a}\langle u\rangle.\,P]\!] \qquad [\![a(x).b(y).P]\!] = [\![b(y).a(x).P]\!]$$

$\rightsquigarrow$ Limits adequacy results . . .

Asynchrony: π-calculus [HT91]

Asynchrony in π-calculus: no continuation after sends.
$\rightsquigarrow$ $\bar{a}\langle u\rangle.\bar{b}\langle v\rangle$ is not a term!

Moreover, in asynchronous π-calculus:

$$a(x).\, b(y).\, P \simeq_{\mathsf{may}} b(y).a(x).Q$$

$\rightsquigarrow$ Models of [Lai05, ST17] **adequate for may**.

Asynchrony: π-calculus [HT91]

Asynchrony in π-calculus: no continuation after sends.
$\rightsquigarrow$ $\bar{a}\langle u\rangle.\bar{b}\langle v\rangle$ is not a term!

Moreover, in asynchronous π-calculus:

$$a(x).\, b(y).\, P \simeq_{\mathrm{may}} b(y).a(x).Q$$

$\rightsquigarrow$ Models of [Lai05, ST17] **adequate for may**.

However,

$$a(x).\, b(y).\, P \not\simeq_{\mathrm{must}} b(y).a(x).Q$$

No adequacy possible for **non-angelic testing equivalences** ...

$\Rightarrow$ Need to take synchrony seriously!

Session types [HVK98]

Typing discpline where **types** are **protocols**:

$$\begin{aligned} S, T ::= {} & \mathsf{end} \\ & \mid \oplus_{i \in I} \ell_i(S_i).\, T_i \\ & \mid \&_{i \in I} \ell_i(S_i).\, T_i \end{aligned}$$

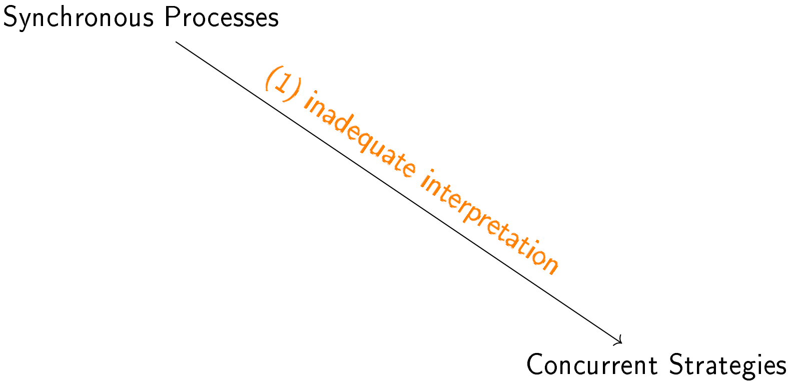
Typing $\vdash P :: a_1 : S_1, \ldots, a_n : S_n$ ensures **protocol preservation**.

$$\frac{\vdash P : a : T_k, \Delta}{\vdash a!\ell_k\langle u\rangle.\, P :: a : \oplus_{i \in I} \ell_i(S_i).\, T_i, \Delta, u : S_k}$$

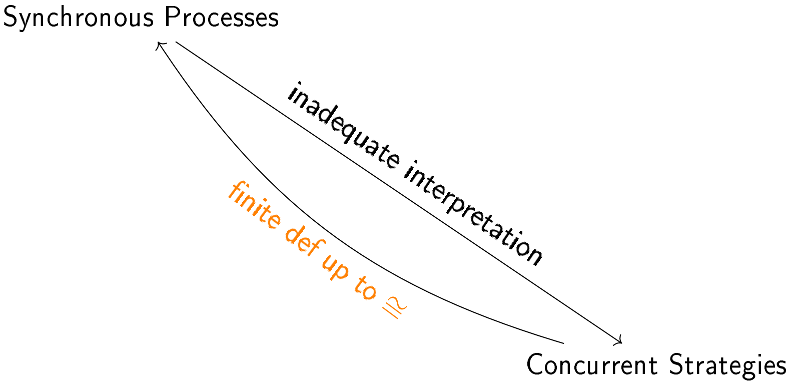
Duality expresses compatible endpoints:

$$\frac{\vdash P :: \Delta, a : S, b : S^{\perp}}{\vdash (\nu ab)\, P :: \Delta}$$

This talk

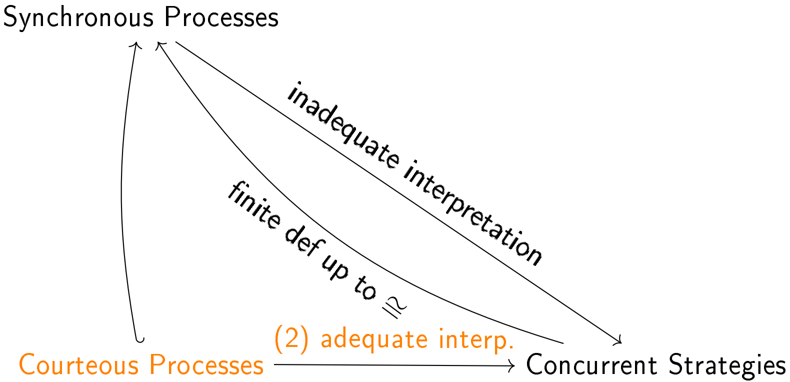

Synchronous Processes
(1) inadequate interpretation
Concurrent Strategies

This talk



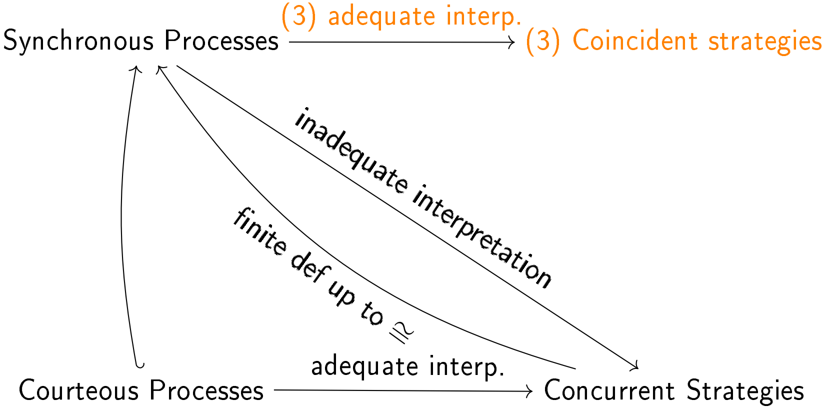
Synchronous Processes
inadequate interpretation
finite def up to ≅
Concurrent Strategies

This talk



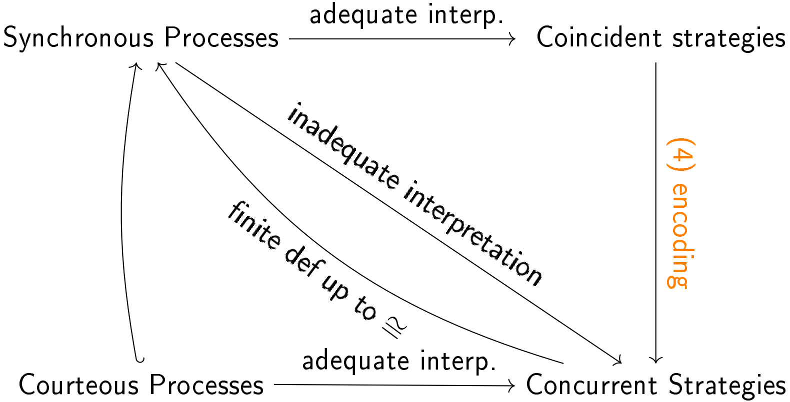
Synchronous Processes
inadequate interpretation
finite def up to ≅
(2) adequate interp.
Courteous Processes
Concurrent Strategies

This talk



Synchronous Processes
(3) adequate interp.
(3) Coincident strategies
inadequate interpretation
finite def up to ≅
adequate interp.
Courteous Processes
Concurrent Strategies

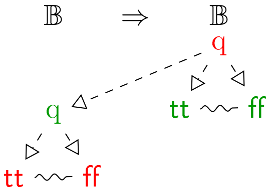
This talk


Synchronous Processes
adequate interp.
Coincident strategies
inadequate interpretation
(4) encoding
finite def up to ≅
Courteous Processes
adequate interp.
Concurrent Strategies

I. Session types into concurrent games

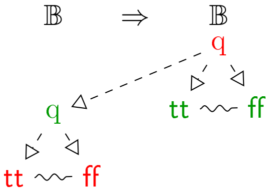
Types as games

In concurrent games, games are **polarized event structures**:



Types as games

In concurrent games, games are **polarized event structures**:



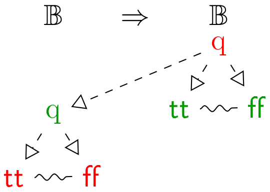
Interpretation of types is given by induction:

$$[\![\&_{i\in I} \ell_i(S_i).T_i]\!] = \sum_{i\in I} \ell_i \cdot ([\![S_i]\!] \parallel [\![T_i]\!])$$

$$[\![\oplus_{i\in I} \ell_i(S_i).T_i]\!] = \sum_{i\in I} \ell_i \cdot ([\![S_i]\!]^{\perp} \parallel [\![T_i]\!])$$

Types as games

In concurrent games, games are **polarized event structures**:



$$= \left[\!\!\left[\begin{array}{c} \&_{i \in \{*\}} i(\oplus_{j \in \{*\}} j(\&_{b \in \{\mathrm{tt},\mathrm{ff}\}} b)). \\ \oplus_{b' \in \{\mathrm{tt},\mathrm{ff}\}} b' \end{array} \right]\!\!\right]$$

Interpretation of types is given by induction:

$$[\![\&_{i \in I} \ell_i(S_i).T_i]\!] = \sum_{i \in I} \ell_i \cdot ([\![S_i]\!] \parallel [\![T_i]\!])$$

$$[\![\oplus_{i \in I} \ell_i(S_i).T_i]\!] = \sum_{i \in I} \ell_i \cdot ([\![S_i]\!]^{\perp} \parallel [\![T_i]\!])$$

Lemma

Every tree-like game is the interpretation of a type.

Processes as strategies

Interpretation is by induction, eg.

$$\left[\!\!\left[\frac{\vdash P : a : T_k, \Delta \qquad k \in I}{\vdash a!\ell_k\langle u\rangle.P :: a : \oplus_{i\in I}\ell_i(S_i).\, T_i, \Delta, u : S_k}\right]\!\!\right] = \ell_k \cdot (\, \mathfrak{C}_{[\![S_k]\!]} \parallel [\![P]\!]).$$

Restriction uses duality:

$$\left[\!\!\left[\frac{\vdash P :: \Delta, a : S, b : S^{\perp}}{\vdash (\nu ab)\, P :: \Delta}\right]\!\!\right] = [\![P]\!] \odot \mathfrak{C}_{[\![S]\!]}.$$

Processes as strategies

Interpretation is by induction, eg.

$$\left[\!\!\left[\frac{\vdash P : a : T_k, \Delta \qquad k \in I}{\vdash a!\ell_k\langle u\rangle.P :: a : \oplus_{i\in I}\ell_i(S_i).\,T_i, \Delta, u : S_k}\right]\!\!\right] = \ell_k \cdot (\,\mathbb{C}_{[\![S_k]\!]} \parallel [\![P]\!]).$$

Restriction uses duality:

$$\left[\!\!\left[\frac{\vdash P :: \Delta, a : S, b : S^{\perp}}{\vdash (\nu ab)\, P :: \Delta}\right]\!\!\right] = [\![P]\!] \odot \mathbb{C}_{[\![S]\!]}.$$

In general $[\![P]\!]$ is **not** courteous, however we still get a sound model:

Lemma

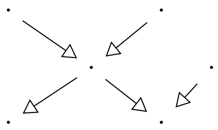
If $P \longrightarrow Q$ then $[\![P]\!] \lesssim [\![Q]\!]$.

Finite definability

Interestingly, if we have any strategy:

$\sigma : \quad [\![\Delta]\!] \qquad\qquad 1 \quad 1 \quad \bot$

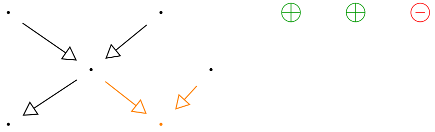
$\oplus \quad \oplus \quad \ominus$



Finite definability

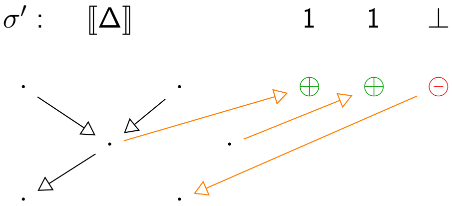
Interestingly, if we have any strategy:

$$\sigma : \quad [\![\Delta]\!] \qquad 1 \quad 1 \quad \bot$$



Finite definability

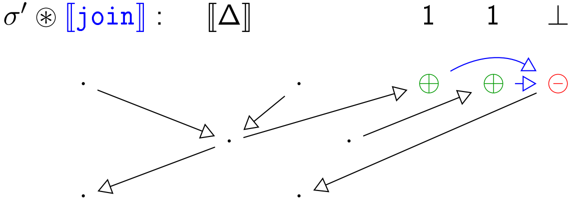
Interestingly, if we have any strategy:



σ' is more tree-like than σ

Finite definability

Interestingly, if we have any strategy:

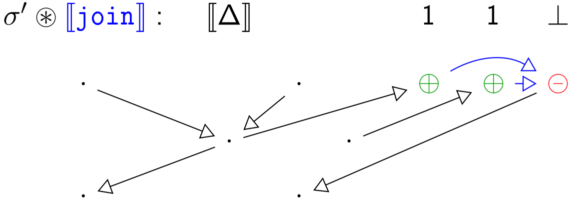


σ' is more tree-like than σ and:

$$\sigma' = \sigma \odot [\![\texttt{join}]\!].$$

Finite definability

Interestingly, if we have any strategy:



σ' is more tree-like than σ and:

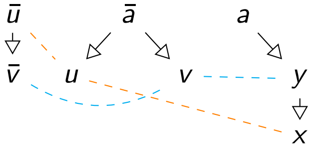
$$\sigma' = \sigma \odot [\![\texttt{join}]\!].$$

Theorem

Every $\sigma : [\![\Delta]\!]$ is the interpretation of a process.

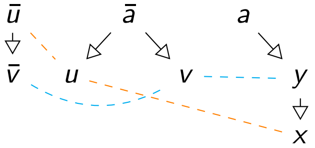
Inadequacy

$(\nu a\bar{a})(\nu u\bar{u})(\nu u'\bar{v}')(\bar{a}\langle u, v\rangle \mid \bar{u}.\bar{v} \mid a(x, y)\, y.x)$ **deadlocks**:

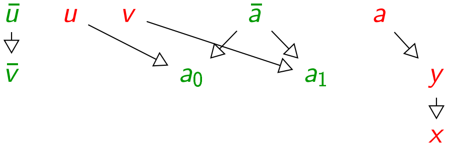


Inadequacy

$(\nu a\bar{a})(\nu u\bar{u})(\nu u'\bar{v}')(\bar{a}\langle u, v\rangle \mid \bar{u}.\bar{v} \mid a(x,y)\, y.x)$ **deadlocks**:



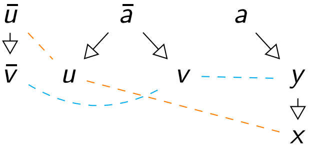
In the model, **copycat** deals with communication and adds delay:



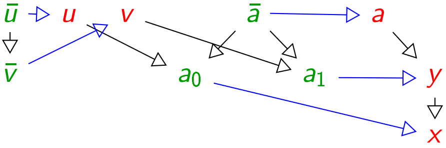
$\rightsquigarrow$ No deadlock anymore.

Inadequacy

$(\nu a\bar{a})(\nu u\bar{u})(\nu u'\bar{v}')(\bar{a}\langle u, v\rangle \mid \bar{u}.\bar{v} \mid a(x, y)\, y.x)$ **deadlocks**:

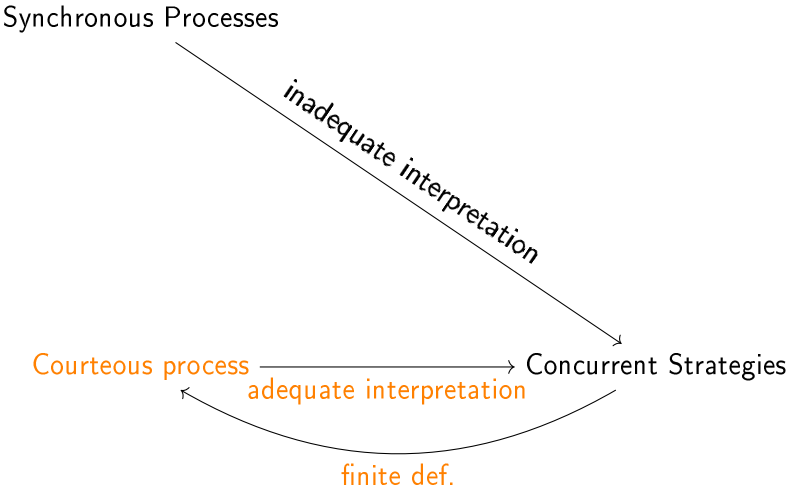


In the model, **copycat** deals with communication and adds delay:



⇝ No deadlock anymore.

II. Courteous processes



Definition & Adequacy

A process P is courteous when $[\![P]\!]$ is courteous.

Lemma

1. *If* $P \longrightarrow Q$ *and* P *is courteous, then* Q *is courteous*
2. *If* $[\![P]\!] \lesssim \tau$ *then* $P \longrightarrow Q$ *with* $[\![Q]\!] = \tau$
3. *Every finite courteous* $\sigma : [\![\Delta]\!]$ *is the interpretation of a courteous* P

A strong link

From these results there is a strong correspondence between:

- The category of session types and courteous processes
- The category of games and strategies of [RW11, CCHW18]

$\rightsquigarrow$ Correspondence seems to play well with bisimulation & obs. eq.

A strong link

From these results there is a strong correspondence between:

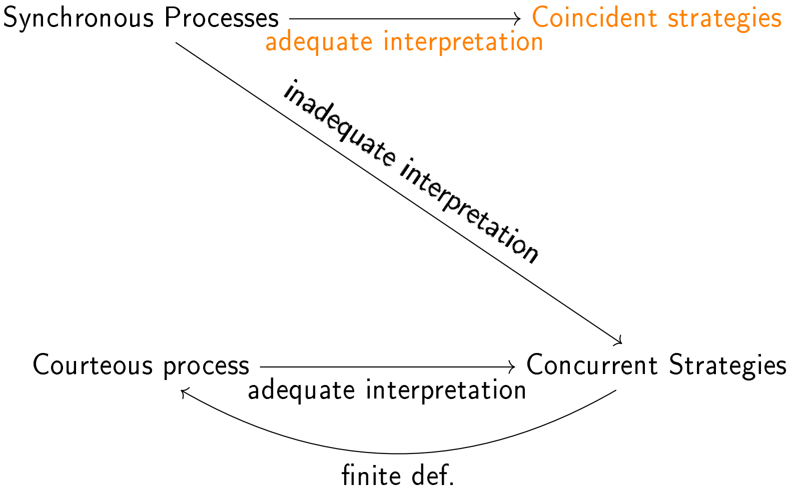
- The category of session types and courteous processes
- The category of games and strategies of [RW11, CCHW18]

$\rightsquigarrow$ Correspondence seems to play well with bisimulation & obs. eq.

Hence:

- Session types and process provide a syntax for strategies
- Equivalent to interpret a language inside one or the other. (Generalizes [HO95] and [BHY01] to true concurrency and non-innocence)

III. Coincident strategies



What is going on

Async forwarder. Given S, there is $\vdash [x = y] :: x : S, y : \bar{S}$ with

$$[\![[x = y]]\!] = \text{\c{c}}_{[\![S]\!]}.$$

What is going on

Async forwarder. Given S, there is $\vdash [x = y] :: x : S, y : \bar{S}$ with

$$[\![[x = y]]\!] = \text{\c{c}}_{[\![S]\!]}.$$

Our model interprets free output **indirectly**, indeed:

$$[\![\bar{a}\langle u \rangle]\!] = [\![(\nu xy)(\bar{a}\langle x \rangle \mid [y = u])]\!].$$

However $(\nu xy)(P(x) \mid [y = u]) \approx P(u)$ *only if P is courteous.*

$\rightsquigarrow$ Change copycat to allow "coincidences" between x and y.

Coincident event structures

In event structures, event occurs separately of the others:

$$\emptyset \subseteq \{a_1\} \subseteq \{a_1, a_2\} \subseteq \ldots.$$

[1] Known as *Completeness* and *Stability*.

Coincident event structures

In event structures, event occurs separately of the others:

$$\emptyset \subseteq \{a_1\} \subseteq \{a_1, a_2\} \subseteq \dots.$$

Definition

A **coincident event structure** is a pair $(E, \mathcal{E})$ satisfying:[1]

- if $x, y \in \mathcal{E}$ bounded in $\mathcal{E}$ then $x \cup y \in \mathcal{E}$ and $x \cap y \in \mathcal{E}$.

Covering chains are not sequences of events but of **coincidences**

$$\emptyset \subseteq X_1 \subseteq X_1 \cup X_2 \subseteq \dots$$

[1] Known as *Completeness* and *Stability*.

Coincident event structures

In event structures, event occurs separately of the others:

$$\emptyset \subseteq \{a_1\} \subseteq \{a_1, a_2\} \subseteq \dots.$$

Definition

A **coincident event structure** is a pair $(E, \mathcal{E})$ satisfying:[1]

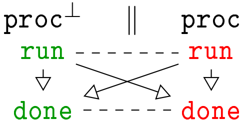
- if $x, y \in \mathcal{E}$ bounded in $\mathcal{E}$ then $x \cup y \in \mathcal{E}$ and $x \cap y \in \mathcal{E}$.

Covering chains are not sequences of events but of **coincidences**

$$\emptyset \subseteq X_1 \subseteq X_1 \cup X_2 \subseteq \dots$$

Given a game A, we can form the **coincident copycat**:

$$\mathrm{ccc_A} = (A^\perp \parallel A, \{x \parallel x \mid x \in \mathscr{C}(A)\})$$



.

[1] Known as *Completeness* and *Stability*.

Coincident strategies

Definition

A coincident strategy on A is a map $\mathcal{S} \to A$ such that its coincidence are singletons or of the form $\{a, b\}$.

$\rightsquigarrow$ A category **without requiring courtesy**!

Coincident strategies

Definition

A coincident strategy on A is a map $\mathcal{S} \to A$ such that its coincidence are singletons or of the form $\{a, b\}$.

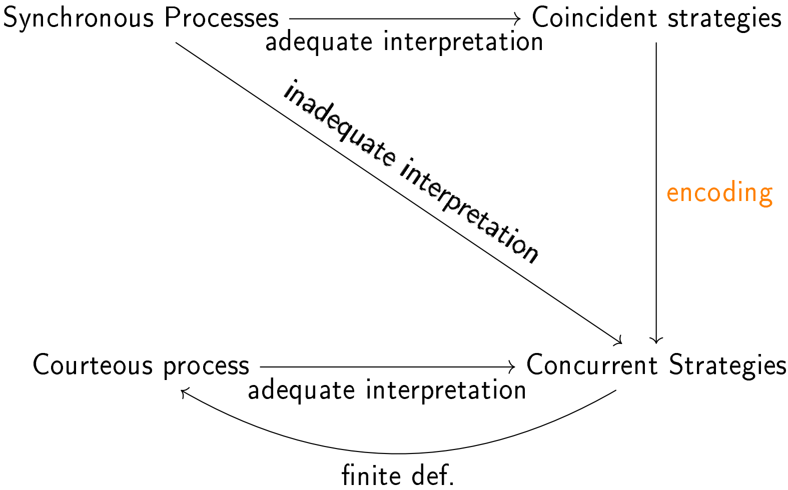
$\rightsquigarrow$ A category **without requiring courtesy**!

We can now change the interpretation of free output:

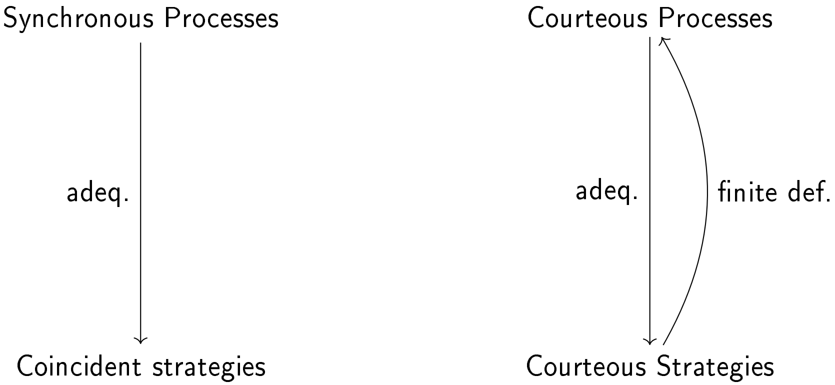
$$\left[\!\!\left[\frac{\vdash P : a : T_k, \Delta \qquad k \in I}{\vdash a!\ell_k\langle u\rangle.P :: a : \oplus_{i\in I}\ell_i(S_i).\, T_i, \Delta, u : S_k}\right]\!\!\right] = \ell_k \cdot (\mathrm{ccc}_{[\![S_k]\!]} \parallel [\![P]\!]).$$

$\rightsquigarrow$ An **adequate interpretation** of synchronous session types.
However: semantic space too broad (no finite definability).

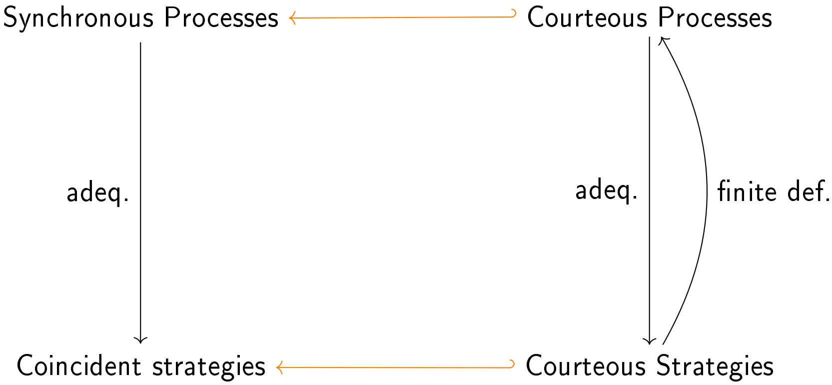
IV. The encoding



Two worlds

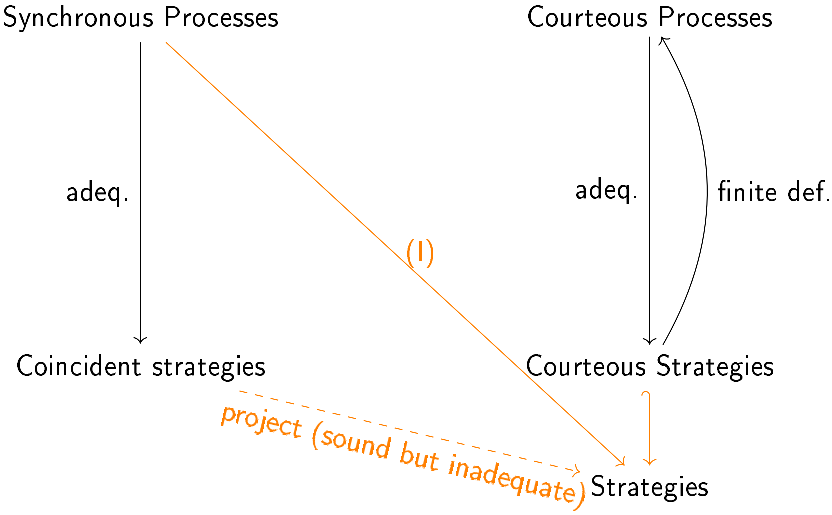

Synchronous Processes
adeq.
Coincident strategies
Courteous Processes
adeq.
finite def.
Courteous Strategies

Two worlds

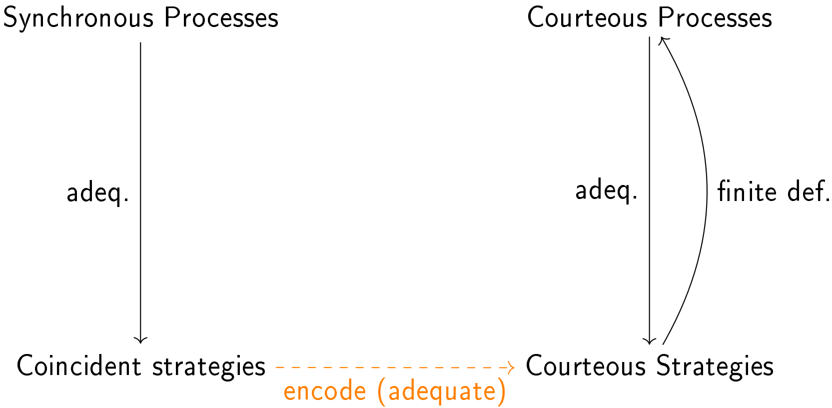


But: diagram does **not** commute

Two worlds

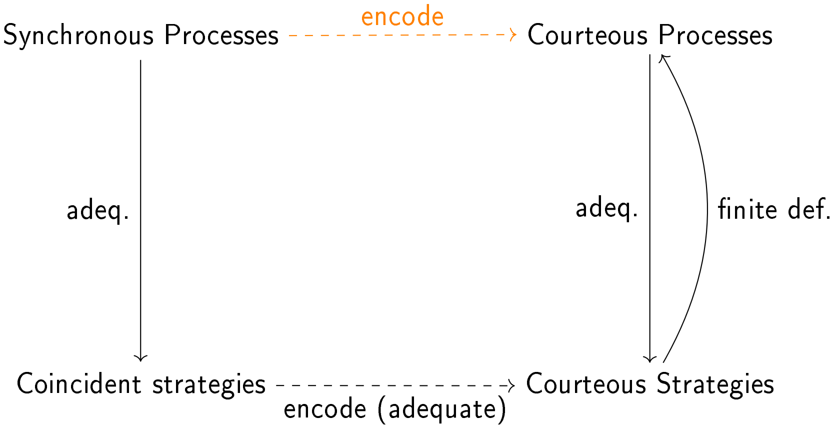

Synchronous Processes
Courteous Processes
adeq.
adeq.
finite def.
(I)
Coincident strategies
Courteous Strategies
project (sound but inadequate)
Strategies

Two worlds



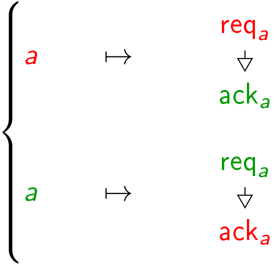
Idea: add **acknowledgements** to protocols

Two worlds

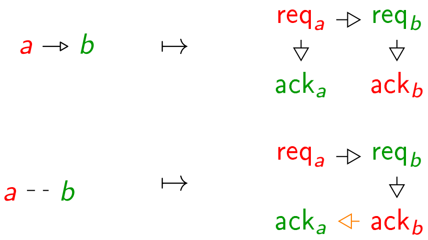

Synchronous Processes
encode
Courteous Processes
adeq.
adeq.
finite def.
Coincident strategies
Courteous Strategies
encode (adequate)

Definition

1. **Unfold the protocol**: $A \mapsto \uparrow A$



2. **Unfold the strategies**: $\sigma \to \uparrow \sigma$



Properties

- Encoding is **injective**:

$$\text{configurations of } \sigma \simeq \textbf{complete} \text{ configurations of } \uparrow \sigma$$

- Should preserve and reflect weak bisimulation

$$\sigma \approx \tau \qquad \text{iff} \uparrow \sigma \approx \uparrow \tau$$

- Characterisation of the image: **well-acknowledging** strategies.

$\rightsquigarrow$ Coincident strategies $\cong$ subcategory of courteous strategies

Summary & Perspectives

- We show a tight correspondance between Session Types and Game Semantics
- Benefits both communities:
  - Provide a precise syntactic description of concurrent strategies
  - Describes the causal behaviour of session processes

Future work.

- Extend to the nonlinear setting.
  $\rightsquigarrow$ A language for innocent concurrent strategies.
- Extend session types to non-tree-like protocols.

Martin Berger, Kohei Honda, and Nobuko Yoshida.
Sequentiality and the pi-calculus.
In *TLCA*, pages 29–45, 2001.

Simon Castellan, Pierre Clairambault, Jonathan Hayman, and Glynn Winskel.
Non-angelic concurrent game semantics.
2018.
Accepted at FOSSACS'18.

Clovis Eberhart, Tom Hirschowitz, and Thomas Seiller.
An intensionally fully-abstract sheaf model for pi.
In Lawrence S. Moss and Pawel Sobocinski, editors, *6th Conference on Algebra and Coalgebra in Computer Science, CALCO 2015, June 24-26, 2015, Nijmegen, The Netherlands*, volume 35 of *LIPIcs*, pages 86–100. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

J. M. E. Hyland and C.-H. Luke Ong.
Pi-calculus, dialogue games and PCF.

In *Proceedings of the seventh international conference on Functional programming languages and computer architecture, FPCA 1995, La Jolla, California, USA, June 25-28, 1995*, pages 96–107, 1995.

Jim Laird.
A game semantics of the asynchronous *pi*-calculus.
In *CONCUR 2005 - Concurrency Theory, 16th International Conference, CONCUR 2005, San Francisco, CA, USA, August 23-26, 2005, Proceedings*, pages 51–65, 2005.

Paul-André Melliès and Samuel Mimram.
Asynchronous games: Innocence without alternation.
In *CONCUR 2007 - Concurrency Theory, 18th International Conference*, pages 395–411, 2007.

Silvain Rideau and Glynn Winskel.
Concurrent strategies.
In *Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science, LICS 2011, June 21-24, 2011, Toronto, Ontario, Canada*, pages 409–418, 2011.

Ken Sakayori and Takeshi Tsukada.
A truly concurrent game model of the asynchronous π-calculus.

In *Proceedings of the 20th International Conference on Foundations of Software Science and Computation Structures - Volume 10203*, pages 389–406, New York, NY, USA, 2017. Springer-Verlag New York, Inc.